



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'jobs.1p' command

\$ man jobs.1p

JOBS(1P) POSIX Programmer's Manual JOBS(1P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

jobs ? display status of jobs in the current session

SYNOPSIS

jobs [-l|-p] [job_id...]

DESCRIPTION

The jobs utility shall display the status of jobs that were started in the current shell environment; see Section 2.12, Shell Execution Environment.

When jobs reports the termination status of a job, the shell shall remove its process ID from the list of those "known in the current shell execution environment"; see Section 2.9.3.1, Examples.

OPTIONS

The jobs utility shall conform to the Base Definitions volume of POSIX.1?2017, Section 12.2, Utility Syntax Guidelines.

The following options shall be supported:

- l (The letter ell.) Provide more information about each job listed. This information shall include the job number, cur?

rent job, process group ID, state, and the command that formed the job.

-p Display only the process IDs for the process group leaders of the selected jobs.

By default, the jobs utility shall display the status of all stopped jobs, running background jobs and all jobs whose status has changed and have not been reported by the shell.

OPERANDS

The following operand shall be supported:

job_id Specifies the jobs for which the status is to be displayed.

If no job_id is given, the status information for all jobs shall be displayed. The format of job_id is described in the Base Definitions volume of POSIX.1?2017, Section 3.204, Job Control Job ID.

STDIN

Not used.

INPUT FILES

None.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of jobs:

LANG Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of POSIX.1?2017, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

LC_ALL If set to a non-empty string value, override the values of all the other internationalization variables.

LC_CTYPE Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).

LC_MESSAGES

Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error

and informative messages written to standard output.

NLSPATH Determine the location of message catalogs for the processing of LC_MESSAGES.

ASYNCHRONOUS EVENTS

Default.

STDOUT

If the -p option is specified, the output shall consist of one line for each process ID:

"%d\n", <process ID>

Otherwise, if the -l option is not specified, the output shall be a series of lines of the form:

"[%d] %c %s %s\n", <job-number>, <current>, <state>, <command>

where the fields shall be as follows:

<current> The character '+' identifies the job that would be used as a

default for the fg or bg utilities; this job can also be specified using the job_id %+ or "%%". The character '-' identifies the job that would become the default if the current default job were to exit; this job can also be specified using the job_id %-. For other jobs, this field is a <space>.

At most one job can be identified with '+' and at most one job can be identified with '-'. If there is any suspended job, then the current job shall be a suspended job. If there are at least two suspended jobs, then the previous job also shall be a suspended job.

<job-number>

A number that can be used to identify the process group to the wait, fg, bg, and kill utilities. Using these utilities, the job can be identified by prefixing the job number with '% '.

<state> One of the following strings (in the POSIX locale):

Running Indicates that the job has not been suspended by a signal and has not exited.

Done Indicates that the job completed and returned exit

status zero.

Done(code)

Indicates that the job completed normally and that it exited with the specified non-zero exit status, code, expressed as a decimal number.

Stopped Indicates that the job was suspended by the SIGTSTP signal.

Stopped (SIGTSTP)

Indicates that the job was suspended by the SIGTSTP signal.

Stopped (SIGSTOP)

Indicates that the job was suspended by the SIGSTOP signal.

Stopped (SIGTTIN)

Indicates that the job was suspended by the SIGTTIN signal.

Stopped (SIGTTOU)

Indicates that the job was suspended by the SIGTTOU signal.

The implementation may substitute the string Suspended in place of Stopped. If the job was terminated by a signal, the format of <state> is unspecified, but it shall be visibly distinct from all of the other <state> formats shown here and shall indicate the name or description of the signal causing the termination.

<command> The associated command that was given to the shell.

If the -l option is specified, a field containing the process group ID shall be inserted before the <state> field. Also, more processes in a process group may be output on separate lines, using only the process ID and <command> fields.

STDERR

The standard error shall be used only for diagnostic messages.

OUTPUT FILES

None.

EXTENDED DESCRIPTION

None.

EXIT STATUS

The following exit values shall be returned:

0 Successful completion.

>0 An error occurred.

CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

APPLICATION USAGE

The `-p` option is the only portable way to find out the process group of a job because different implementations have different strategies for defining the process group of the job. Usage such as `$(jobs -p) pro?` provides a way of referring to the process group of the job in an implementation-independent way.

The `jobs` utility does not work as expected when it is operating in its own utility execution environment because that environment has no applicable `jobs` to manipulate. See the APPLICATION USAGE section for `bg`.

For this reason, `jobs` is generally implemented as a shell regular built-in.

EXAMPLES

None.

RATIONALE

Both `%%` and `%+` are used to refer to the current job. Both forms are of equal validity; the `%%` mirroring `$$` and `%+` mirroring the output of `jobs`. Both forms reflect historical practice of the KornShell and the C shell with job control.

The job control features provided by `bg`, `fg`, and `jobs` are based on the KornShell. The standard developers examined the characteristics of the C shell versions of these utilities and found that differences exist.

Despite widespread use of the C shell, the KornShell versions were selected for this volume of POSIX.1:2017 to maintain a degree of uniformity.

mity with the rest of the KornShell features selected (such as the very popular command line editing features).

The jobs utility is not dependent on the job control option, as are the seemingly related bg and fg utilities because jobs is useful for examining background jobs, regardless of the condition of job control. When the user has invoked a set +m command and job control has been turned off, jobs can still be used to examine the background jobs associated with that current session. Similarly, kill can then be used to kill background jobs with kill %<background job number>.

The output for terminated jobs is left unspecified to accommodate various historical systems. The following formats have been witnessed:

1. Killed(signal name)
2. signal name
3. signal name(core dump)
4. signal description- core dumped

Most users should be able to understand these formats, although it means that applications have trouble parsing them.

The calculation of job IDs was not described since this would suggest an implementation, which may impose unnecessary restrictions.

In an early proposal, a -n option was included to "Display the status of jobs that have changed, exited, or stopped since the last status report". It was removed because the shell always writes any changed status of jobs before each prompt.

FUTURE DIRECTIONS

None.

SEE ALSO

Section 2.12, Shell Execution Environment, bg, fg, kill, wait

The Base Definitions volume of POSIX.1-2017, Section 3.204, Job Control Job ID, Chapter 8, Environment Variables, Section 12.2, Utility Syntax Guidelines

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portion of IEEE Std 1003.1-2017

table Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html .

IEEE/The Open Group

2017

JOBS(1P)